**Session 1: 1\_1 Getting your data.R**  
(1) Import the text file data file "TMS\_data.txt" to R Studio and store it in an object named “Mydata”

Mydata <- read.table(file="TMS\_data.txt",header=TRUE)

(2) What is the name of the 3rd column?  
  
names(Mydata)   
names(Mydata)[3]  
  
Answer: Block  
  
(3) What is the value of the 15th row in the 5th column?  
  
Mydata[15,5]

Mydata$ACC[15]

Answer: 1

(4) Go to http://www.statmethods.net/input/contents.html

Choose another command to apply to Mydata – what does it do?

(5) Generate a histogram, density plot and qq/quantile plots for the Twitches column in Mydata, Make sure they all display on the same window

par(mfrow=c(2,2))

hist(Mydata$Twitches)

plot(density(Mydata$Twitches,na.rm=TRUE))

qqnorm(Mydata$Twitches)

qqline(Mydata$Twitches)

(6) Generate a table / output of descriptive statistics for the Reaction Time data in Mydata, for each participant/subject. Store it in an object called ‘Subject.dat’  
Make sure the output is in a matrix/dataframe (not a list)

Subject.dat<-describeBy(Mydata$RT,group=Mydata$SubNo,mat=TRUE)

(7) What is the mean and standard deviation RT for Subject Number 4?

Mean = 0.33 SD = 0.039

**Session 2: 1\_2 Manipulating Data.R**

(1) Filter data from Mydata, for all Congruent conditions on the CRT task

filter(Mydata, Congruence == "Cong", Task == "CRT")

(2) Generate summary data for the Reaction Time in milliseconds (RTms) for each participant, for each level of Axes. For each participant, you want the mean RT, the standard deviation, the minimum and maximum.  
Store the grouped data in Participant.Axes

Participant.Axes <- group\_by(Mydata, SubNo, Axes)

# summary RT

Participant.Axes <- summarise(Participant.Axes,

MeanRT = mean(RTms),

SDRT = sd(RTms),

minimum = min(RTms),

maximum = max(RTms))

First few rows should look like this:

# A tibble: 80 x 6

# Groups: SubNo [?]

SubNo Axes MeanRT SDRT minimum maximum

<int> <fctr> <dbl> <dbl> <dbl> <dbl>

1 1 E.W 434.7356 63.52317 293.305 647.261

2 1 N.S 509.4579 321.54244 308.097 2000.232

3 1 NE.SW 436.3371 77.00437 304.647 680.271

4 1 NW.SE 433.2976 84.70414 258.502 841.720

5 2 E.W 427.1174 96.05200 292.408 752.216

6 2 N.S 421.6929 113.86346 281.399 1269.995

7 2 NE.SW 421.5957 110.91452 276.681 825.877

(3) Create new data using Mydata that adds 5 to each value of Twitches.   
Do not store this in Mydata. Name the new column ‘Twitch5’

mutate(Mydata, Twitch5 = Twitches+5)

(4) Repeat the dcast function, but this time summarise the data in ‘temp’ so that there is one column for each level of Twitches.

dcast(temp, RT\_variable + SubNo ~ Twitches,

+ fun.aggregate = mean, na.rm = TRUE)

(5a) Extract the following from ‘Mydata’: the columns for SubNo, Congruence, RT

Store this in a data object called ‘CongTemp’

CongTemp <- select(Mydata, SubNo, RT, Congruence)

(5b) How many rows and columns does CongTemp have?  
Rows = 9600, columns = 3

dim(CongTemp)

9600 3

(6) use melt and cast to summarise the data so that you have one column for each level of Congruence, with the average RT for each participant.  
When you ‘melt’ the data, name the variable “DataType” and the value “RT\_value”.

temp <- melt(CongTemp, id.vars = c("SubNo", "Congruence"),

variable.name = "DataType",

value.name = "RT\_value")

then

dcast(temp, DataType + SubNo ~ Congruence,

fun.aggregate = mean, na.rm = TRUE)

(7) Relabel the CongTemp$Congruence column so that   
Incong = Incongruent and   
Cong = Congruent

To do this, overwrite the data in CongTemp$Congruence

CongTemp$Congruence<-revalue(CongTemp$Congruence, c("Incong" = "Incongruent", "Cong" = "Congruent"))

(8) Recreate the messy data we used before (with Haircut and Biscuits columns)  
Trim the data in the Column HairCut to remove spaces. Overwrite the original data.

In the column Haricut replace the cells called ‘hair’ with NA

x<-c(" hair", "hair", " nohair", " ", "hair ", " ", "nohair", "hair ", "XXX", " nohair", "CCC")

y<-sample(1:20, 11)

x<-data.frame(cbind(x,y))

names(x)<-c("HairCut","Biscuits")

x$HairCut <-trim(x$HairCut)

x$HairCut[which(x$HairCut=="hair")] <- NA

*Should look like this…*

HairCut Biscuits

1 <NA> 9

2 <NA> 16

3 nohair 20

4 1

5 <NA> 18

6 4

7 nohair 3

8 <NA> 11

9 XXX 13

10 nohair 6

11 CCC 14

**Session 3: 2\_1 Normality\_Correlation\_TwoSampleTests.R**

(1) Go back to the first script 1\_1Getting your data.R

Find the code that plots a histogram of data and overlays a normal curve

Adapt this code to plot the CongTemp$RT data and overlay a normal curve.

Lines 255-267

#create some data that is normally distributed, using the mean and sd from our RT data

d<-na.omit(CongTemp$RT) #Remove NAs from data so R doesn't get upset

dnorm(seq(min(d),max(d),length=100),mean=mean(d),sd=sd(d))

hist(d,freq=FALSE) #plot histogram with probability not frequency counts

#create normal distribution data and add to plot

#make sure to highlight both lines of code and run them

points(seq(min(d),max(d),length=100),

dnorm(seq(min(d),max(d),length=100),mean=mean(d),sd=sd(d)),type="l",col="red")

*Note that the ONLY thing you had to change here was the initial piece of code for how the data object ‘d’ is generated. Much of the work in R is intelligent copy paste.*

(2) Using the data from CongTemp, correlate the RTs for each participant in the Congruent and Incongruent conditions.  
First, you will need to use melt/case (reshape2) to put the RTs into separate columns (see code on lines 175-204 in 1\_2Manipulating Data.R).  
Save your melted/casted data into an object called ‘x’  
Then check normality. If data is normally distributed, use pearson. If it is not, use spearman.  
Use a scatterplot to see the correlation data

x <- melt(CongTemp, id.vars = c("SubNo", "Congruence"),

variable.name = "RT\_variable",

value.name = "RT\_value")

head(x)

x <- dcast(x, RT\_variable + SubNo ~ Congruence,

fun.aggregate = mean, na.rm = TRUE)

head(x)

hist(x$Congruent)

hist(x$Incongruent)

qqnorm(x$Congruent)

qqline(x$Congruent)

cor(x$Congruent, x$Incongruent, method="spearman", use="pairwise.complete.obs")

plot(x$Congruent, x$Incongruent)

(3) Complete Wilcox test on the data in ‘congruence’

wilcox.test(congruence$Cong,congruence$Incong,paired=TRUE)

(4) Complete a Wilcoxon test on the data in ‘Task’.  
Create a boxplot of the data in ‘Task’

wilcox.test(Task$RTms~Task$Experiment)

boxplot(Task$RTms ~ Task$Experiment)

**Session 4: 2\_2 ANOVA.R**

(1) Another way to transform RT data is to use an inverse transform (i.e. 1/RT).

Take the RTms data in Mydata, and create a new column with the inverse transformed data. Call this column InvRT  
   
Look at this data in a histogram and with quantile plots. Does it improve the distribution?

Mydata.corr$InvRT <- 1/Mydata.corr$RTms

hist(Mydata.corr$InvRT)

qqnorm(Mydata.corr$InvRT)

qqline(Mydata.corr$InvRT)

(2) Create a boxplot of the inverse transformed RT data. From this, create a variable called ‘z’ that stores the outliers.

mybp <- boxplot(InvRT ~ SubNo, data=Mydata.corr)

z<-which(Mydata.corr$InvRT %in% mybp$out, arr.in=TRUE)

(3) Finally, create a column for these outliers called ‘OutInv’. In that column, mark all the rows which show outliers on the boxplot. Tip: pay attention to exactly where you need to change the original code.

Mydata.corr$OutInv<-rep("FALSE")

#loop to identify rows and put 'TRUE' for each outlier

for (i in 1:length(z))

{Mydata.corr$OutInv[z[i]]<-"TRUE"}

#make new column a factor/category

Mydata.corr$OutInv<-as.factor(Mydata.corr$OutInv)

head(Mydata.corr)

(4) Which participants have the most outliers for inverse transformed RT data?

Which participants have fewest outliers?   
How much data is removed if you take out outliers for inverse transformed data?

table(Mydata.corr$SubNo,Mydata.corr$OutInv)  
*Participant 5 and 20 have the most with 13  
Participant 2 has the least with 1*

temp<-filter(Mydata.corr,OutInv==FALSE)

1-dim(temp)/dim(Mydata.corr)  
*It is about 1.5%*

(5) Go back to Mydata.corr.a (this the data with errors and outliers removed).  
Extract summary data for each participant, with their mean RT for each level of HomLocation.   
Remember to rename the columns in your new set, so they are not called ‘variable’ and ‘value’  
What is the mean RT for the scalp location PZ? (across all subjects)

temp<-dcast(Mydata.corr.a, SubNo ~ HomLocation, value.var = "RTms",

fun.aggregate = mean, na.rm = TRUE)

data.summary<-melt(temp, id.vars = "SubNo")

names(data.summary)[2:3]<-c("HomLocation","meanRT")

with(data.summary, tapply(meanRT, HomLocation, mean))

*PZ Mean RT = 443.14*

(6) Complete a 1-way repeated measures ANOVA for HomLocation.  
HomLocation was manipulated within subjects.  
Use the log of mean RT.  
Is there a significant main effect of HomLocation (scalp location)?

aov.1 <-aov(log(meanRT) ~ HomLocation + Error(SubNo/HomLocation), data.summary)

*No significant main effect (F(23,432) = 0.17, p>1)*

(7) Complete follow up pairwise t-tests. Use the Holm method to control for false positives.  
Is there a significant difference between scalp locations ATL and CZ?  
Is there a significant difference between scalp locations CP1.2 and F7.8

pairwise.t.test(log(data.summary$meanRT), data.summary$HomLocation, p.adjust.method="holm", paired=T)

*ATL and CZ: p = 0.00079 Yes  
CP1.2 and F7.8 = 0.01030 Yes*

(8) To look at the data, create a boxplot of the RTs by each HomLocation

par(mfrow=c(1,1))

boxplot(data.summary$meanRT ~ data.summary$HomLocation)

**Session 5: 2\_3 More on ANOVA.R**

(1) Complete an ANOVA analysis with Axes, TwitchFactor and Task (2x1 Mixed ANOVA).  
In this analysis, look for the interaction and main effects for Axes x TwitchFactor, but only look at the main effect for Task.  
Store the result of this analysis in ‘aov.4’  
What are the results?  
  
aov.4 <- aov(log(RTms) ~ (Axes\*TwitchFactor) + Task

+ Error(SubNo/(Axes\*TwitchFactor)), data=Mydata.corr.a)

*Significant main effect of TwitchFactor  
Significant main effect of Task  
No effect of Axes  
No interaction between Axes and TwitchFactor*

(2) Complete an ANOVA analysis with TwitchFactor and Hemlabel.   
Look only at the interaction between these two variables.

Store the result of this analysis in ‘aov.4’  
Plot the interaction between these two variables  
What are the results? Use the plot to provide an interpretation (i.e. explain what the results mean)

aov.4 <- aov(log(RTms) ~ TwitchFactor:Hemlabel

+ Error(SubNo/TwitchFactor), data=Mydata.corr.a)

*with(Mydata.corr.a, interaction.plot(TwitchFactor, Hemlabel, log(RTms), fun = mean))*

*Significant interaction between TwitchFactor and Hemlabel*

*For the right hemisphere, not much difference between high and low twitches.  
For the left hemisphere, higher reaction times for High twitches, and shorter reaction times for Low twitches*

(3) Repeat the ANOVA above, with Axes, TwitchFactor and Task (2x1 Mixed ANOVA). Store it in ‘aov.4’  
Compare the output for this ANOVA with Type II and Type III sums of squares.  
Use contr.sum for the contrasts

aov.4 <- lm(log(RTms) ~ (Axes\*TwitchFactor) + Task

+ Error(SubNo/(Axes\*TwitchFactor)), data=Mydata.corr.a,  
 contrasts = list(Axes = contr.sum, TwitchFactor = contr.sum)

(4) Generate summary data in preparation for a between subjects ANOVA.

We want to look at the effects of Task and Hemisphere (Hemlabel)

To generate the summary data, use the ‘summarise’ function from dplyr.  
Code for this is on Script 2\_1, lines 42-51

Participant.Mydata <- group\_by(Mydata.corr.a,SubNo,Task,Hemlabel)

Participant.Mydata <- summarise(Participant.Mydata,

MeanRT = mean(RT),

SDRT = sd(RT))

(5) Using that summary data, complete a between subjects ANOVA to look at the effects of Task and Hemisphere (Hemlabel) on log RT, using ezANOVA.  
Save this in ‘bet.anova’. Use Type III.  
Plot the data using interaction.plot.

What are the results?  
What is the mean RT for Left hemisphere in the Flanker task?

bet.anova <- ezANOVA(data=Participant.Mydata,   
 dv = log(MeanRT), wid = SubNo,

between = .(Task,Hemlabel),

type=3)

with(Participant.Mydata, interaction.plot(Task, Hemlabel, log(MeanRT), fun = mean))

summary.anova <- ezStats(data=Participant.Mydata,   
 dv = log(MeanRT), wid = SubNo,

between = .(Task,Hemlabel))

*Significant effect of Task, no effect of Hemisphere and no interaction.*

*Mean RT for Left hemisphere in the Flanker task = 0.55 log(ms)*

(6) Complete a mixed ANOVA using the data Mydata.corr.a  
Test the effects of the variables Task (between subjects) and HomLocation (within subjects).  
Write code to complete this ANOVA using both ezANOVA Type 3 *and*   
the afex function aov\_car  
Use lsmeans to get the complete set of means for each cell in the design  
How many levels / conditions are there in HomLocation?   
How many total cells are there?  
Complete a post-hoc test to compare the mean RT for the ATL in the CRT task to CZ in the CRT task. What is the result?

*ezANOVA:*

Participant.Mydata <- group\_by(Mydata.corr.a,SubNo,Task,HomLocation)

Participant.Mydata <- summarise(Participant.Mydata,

MeanRT = mean(RT),

SDRT = sd(RT))

mix.anova <- ezANOVA(data=Participant.Mydata, dv = log(MeanRT), wid = SubNo,

within = .(HomLocation),

between = .(Task),

type=3,

return\_aov = TRUE)

*afex aov\_car:*

mix.anova <- aov\_car(log(MeanRT) ~ (Task\*HomLocation) + Error(SubNo/(HomLocation)),

data=Participant.Mydata, return = "aov")

(ref1 <- lsmeans(mix.anova, c("Task", "HomLocation")))

unique(Participant.Mydata$HomLocation)

unique(Participant.Mydata$Task)

*There are 24 levels/conditions in HomLocation, and two levels of Task.   
This gives 48 cells.*

x<-rep(0,48)

x[3]<-1

x[11]<- -1

c\_list <- list("CRT.ATL - CRT.CZ" = x)

summary(contrast(ref1, c\_list), adjust = "holm") *OR*

summary(as.glht(contrast(ref1, c\_list)), test = adjusted("bonferroni"))

This comparison is significant

**Session 6: 2\_4 Regression.R**

(1) Using what you learnt from ANOVA models, write some code for a regression that tests the main effects between Task and Congruence. Save in lm.2

lm.2<-lm(log(RTms) ~ Task + Congruence, data=Mydata.corr.a)

(2) Write some code for a regression that tests the main effects of Twitches, Task and Congruence. Use log of RTms. Save in lm.3

lm.3<-lm(log(RTms) ~ Twitches + Task + Congruence, data=Mydata.corr.a)

(3) Write some code for a regression that tests the main effect of Twitches, and the main effects and interaction between Task and Congruence. Save in lm.4

lm.4<-lm(log(RTms) ~ Twitches + (Task\*Congruence), data=Mydata.corr.a)

(4) Write some code for a regression that tests the main effect of Twitches, and ONLY the interaction between Task and Congruence. Save in lm.5

lm.5<-lm(log(RTms) ~ Twitches + Task: Congruence, data=Mydata.corr.a)

(5) Make sure you have run all your models. Look at the plots for each one – do the residuals look OK / normally distributed? Make sure you use plot and hist(resid()).  
  
plot(lm.2)  
hist(resid(lm.2))

*lm.2 – residuals are ‘binned’ –this is because we have basically done an ANOVA with 2x2 (Task has two conditions, Congruence has two conditions). Histogram looks OK.  
  
lm.3 – bit better spread but still binned. Histogram looks OK.  
  
lm.4 – identical to lm.3.  
lm.5 – identical to lm.3 and lm.4*

(6) How can we solve the problem that lm.2 and lm.3 are fit to different size data sets? How might we approach this problem? Talk to your neighbours to problem solve this.

*We can reduce the data set that lm.2 is fit with, so that it matches lm.3. For this, we need to take only the complete cases i.e. filter the data that we use, so that it already excludes the missing data we have in Twitches.*

*dim(Mydata.corr.a)*

*dim(Mydata.corr.a[complete.cases(Mydata.corr.a),])*

*temp <- Mydata.corr.a[complete.cases(Mydata.corr.a),]*

lm.2<-lm(log(RTms) ~ Task + Congruence, data=temp)

*anova(lm.2,lm.3)*

(7) Based on the comparisons we have made, which is the best model?  
Based on this model, what combination of predictors best explain the variation in RT?

*lm.4 – the interaction is marginal, so could argue for keeping it in.*

*Main effect of Twitches, and main effects and interactions for Task and Congruence.*

**Session 7: 3\_1 Introduction to LMMs.R**

(1) Build an LMM that includes fixed effects for Trial, main effects for Twitches and main effects and interactions between Task and Congruence and main effect of Twitches.  
Include random slopes for Twitches varying across Subjects, and random slopes for the effect of Congruence varying across Subjects. Include correlations between intercepts and slopes.  
Save this in lmer.4

lmer.4<-lmer(log(RTms) ~ Trial + Twitches + (Task \* Congruence)

+ (1 + Twitches|SubNo) + (1+Congruence|SubNo), data = Mydata.corr.a)

(2) Plot the fixed effects from this model.

eff<-allEffects(lmer.4)

plot(allEffects(lmer.4))

(3) Use update( ) to compare this model to a model that does not include the interaction between Task and Congruence. Save this model in lmer.4.NoInt  
What is the result?  
  
lmer.4<-lmer(log(RTms) ~ Trial + Twitches + (Task \* Congruence)

+ (1 + Twitches|SubNo) + (1+Congruence|SubNo), data = Mydata.corr.a)

lmer.4.NoInt <- update(lmer.2,.~.-Task:Congruence)

(4) Plot the random effects for Congruence varying across Subjects. Make sure you update the axes labels.

lattice::xyplot(fitted(lmer.4) ~ Congruence|SubNo, data=Mydata.corr.a,

main="Congruence by Subject Random Effects",

ylab="Fitted log RT",xlab="Congruence")

(4) Using the MuMIN package, get the R2 value for lmer.4

lmer.0<-lmer(log(RTms) ~ (1 + Twitches|SubNo) + (1+Congruence|SubNo),   
 data = Mydata.corr.a, REML = FALSE)

lmer.4<-lmer(log(RTms) ~ Trial + Twitches + (Task \* Congruence)

+ (1 + Twitches|SubNo) + (1+Congruence|SubNo), data = Mydata.corr.a, REML = FALSE)

r.squaredLR(lmer.4, null=lmer.0)

**Session 7: 3\_2 Graphs.R**

(1) Add axes labels and a title to the graph of Twitches x Task we just created

# add axis labels

plot<-plot + xlab("Twitch Rating")

plot<-plot + ylab("Reaction Time (ms)")

plot

# add title

plot<-plot + ggtitle("Task x Twitch Rating")

plot

(2) Create a bar plot that shows the RT data by each Task. Include error bars that are 1 Standard Error. Add titles and axis labels

Participant.Mydata <- group\_by(Mydata.corr.a,SubNo,Task)

Participant.Mydata <- summarise(Participant.Mydata,

SubjRT = mean(RT),

SubjSDRT = sd(RT))

Participant.Mydata <- group\_by(Participant.Mydata,Task)

data.summary <- summarise(Participant.Mydata,

Mean = mean(SubjRT),

StErr = mean(SubjSDRT)/sqrt(20))

plot <- ggplot(data.summary,aes(x=as.factor(Task),y=Mean))

plot <- plot + geom\_bar(stat="identity")

plot <- plot + geom\_errorbar(aes(ymin=Mean-(StErr),ymax=Mean+(StErr),width=.2))

plot<-plot + xlab("Task")

plot<-plot + ylab("Reaction Time (ms)")

plot<-plot + ggtitle("RT by Task")

(3) Using the code from the webpage, create a scatterplot for Reaction Time by Twitches. Add a regression line.

ggplot(Participant.Mydata, aes(x=Twitches, y=SubjRT)) +

+ geom\_point(shape=1) + geom\_smooth(method=lm)